Simulation of Equipment Design Optimization in Microelectronics Manufacturing

C. H. Tong and J. C. Meza

Scientific Computing Department
Sandia National Laboratories
Livermore, CA 94551-0969
chtong@ca.sandia.gov
meza@ca.sandia.gov

C. D. Moen

Thermal & Plasma Processes Department
Sandia National Laboratories
Livermore, CA 94551-0969
cmoen@ca.sandia.gov

Abstract

We explore different mathematical formulations, develop an object-oriented simulation environment, and perform a parametric study in the design of a high-yield, low-cost (raw material), and short-cycle-time chemical vapor deposition (CVD) reactor for microelectronic manufacturing. We begin with several possible configurations for the reactor and formulate them into their corresponding numerical optimization problems. We then develop a software architecture for solving the optimization problems by integrating the heat conduction and species transport simulation codes and a modern optimization software into an object-oriented optimization environment. Numerical experiments are performed, reported, and discussed.

1 Introduction

Chemical vapor deposition (CVD) techniques have been widely used in the semiconductor industry for integrated circuit fabrication. Complex circuits consisting of thousands to millions of transistors are formed on silicon wafers by repeated sequences of CVD, masking, and etching steps creating a number of superimposed layers of conducting, insulating, and transistor-forming materials. As device feature-scales continue to decrease below the sub-micron level, precision control of these processing steps becomes extremely important to ensure high production yield. Computational modeling and simulation have been demonstrated to be valuable tools in understanding CVD processes and improving reactor and process designs [1, 2, 3].

Of interest in our present study is the deposition uniformity of a CVD process, i.e. variation in the thicknesses of materials formed on various parts of the wafer surface as a result of controlled chemical reactions. A design goal is therefore to predict the "control" parameters for a steady-state deposition phase with a prescribed uniformity requirement. Other design goals are, for example, low raw material cost and short cycle time. For the reactor/process models considered in this work, the controls in the CVD reactor are the reactor wall temperatures and the flow rate of the chemical species injector. We also consider more sophisticated models with multiple injectors. Other more interesting but even more complex models may look at injection rate/temperature time schedule to optimize integrated thickness, but they are not reported here.

A number of nonlinear minimization problems arise from defining different design goals for the deposition process. The minimization problems are solved by using optimization software called NPSOL developed by Gill, Murray, Saunders, and Wright [4]. The simulation code is a program called OVEND developed by Grcar and Houf [5] which computes deposition rate distribution on the wafers given the reactor characteristics and operating conditions. These two programs are integrated into an object-oriented environment through an interface in the OPT++ optimization library (an object-oriented library developed by Meza [6]).

In Section 2, we derive several formulations for the optimal design problem and study their computational needs. In Section 3, we describe a software architecture for the solution of the optimal design problem. In Section 4, we present and discuss numerical results of a linear temperature profile problem.

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

2 The CVD Reactor Design Problem

We consider a small-batch, fast ramp (SBFR) furnace [7], which is designed to heat up and cool down quickly, thus reducing cycle time and thermal budget. A typical SBFR consists of 50 eight-inch (diameter) equally-spaced silicon wafers enclosed in a vacuum-bearing quartz jar (Figure 1). The high operating temperatures (about $900 - 1000$ K) are generated by resistive coil heaters contained in an insulated canister along the length of the reactor. The heaters can be individually controlled to give piecewise linear temperature profiles. In this paper, however, we only consider ganging the heaters together to manifest a linear (with respect to distance from the bottom of the wafer stack) temperature profile on the reactor wall which can be represented by temperatures at two selected positions along the reactor wall. One or more injectors can be placed in the reactor to introduce chemical species. The injectors are characterized by their positions, temperatures, flow rates, and chemical species compositions. This reactor is simulated by an analysis code, OVEND, which consists of a heat conduction model to describe the temperature distribution in the reactor, a species transport model to compute the composition of species inside the reactor [8, 9], and finally a surface kinetics model [10] to describe the growth rate of the deposition.

High production yield, low raw material cost, and short cycle time are important characteristics in CVD process design. If we cast these requirements into an optimal design problem, the objectives are : to maximize the mean deposition rate (to reduce cycle time), to minimize the injector flow rate (to reduce raw material costs), and to minimize the deposition non-uniformity (to improve yield). Since these conflicting requirements may make the simultaneous optimization of these three quantities unattainable, we instead examine several simplifications which are presented in the next few subsections. The following notations are used throughout this report :

$\boldsymbol{T}$	- an array of n wall te mperatures
$\dot{Q}$	- injector flow rate (for 1-injector case)
$\dot{Q}_i$	- flow rate of the i-th injector
$\dot{\boldsymbol{\sigma}}$	- deposition rates at selected locations
$\bar{\dot{\sigma}}$	- mean deposition rate.

2.1 Minimizing Non-uniformity Using Single Injector and a Fixed $\bar{\dot{\sigma}}$

Suppose we have prescribed a uniform target deposition rate $\dot{\sigma}^*$, and we are given one injector at a fixed location (say, 9.75 centimeters from the base of the reactor). The design objective is to find the wall temperature profile and the injector flow rate such that the actual deposition rates at all grid points are as close to $\dot{\sigma}^*$ as possible (based on, for example, the L_2-norm metric). Suppose in addition that the wall temperatures and the flow rate of the injector are constrained to fall within a certain range of values (for example, $948 - 1023$ K for temperatures and $25 - 125$ standard cubic centimeters per minute for the flow rate). The objective function is defined by a least-squares fit of the m discrete deposition rates to the target rate subject to some temperature and flow rate constraints,

$$F(\boldsymbol{T}, \dot{Q}) = \min_{\boldsymbol{T}, \dot{Q}} \frac{1}{m} \sum_{i=1}^{m} (\dot{\boldsymbol{\sigma}}_i - \dot{\sigma}^*)^2 \qquad (1)$$

subject to

$$T_L \leq \boldsymbol{T}_i \leq T_U, \quad i = 1, 2, \cdots, n;$$

$$Q_L \leq \dot{Q} \leq Q_U;$$

where T_L and T_U (Q_L and Q_U) are the lower and upper bounds for the wall temperatures (flow rate), respectively. This bound-constrained, nonlinear, least-squares problem can be solved, for example, by a bound-constrained Gauss-Newton method.

2.2 Maximizing Mean Deposition Rate Using One injector

The best target deposition rate to select is not actually known a priori. A desired capability to be used with OVEND is to predict the maximum allowable (average) deposition rate $\bar{\dot{\sigma}}$ while using minimum amount of chemical species (i.e. minimum flow rate) and such that the deposition rate variation across the wafers is kept at a minimum. A more practical formulation is to relax the minimum flow rate and deposition uniformity requirements. In particular, it is satisfactory to allow the deposition rate non-uniformity to fall within a user-specified relative tolerance. Let $\tau > 0$ be a user-defined absolute tolerance. Then the problem can be formulated as

$$F(\boldsymbol{T}, \dot{Q}) = \min_{\boldsymbol{T}, \dot{Q}} -\frac{1}{m} \sum_{i=1}^{m} \dot{\boldsymbol{\sigma}}_i \qquad (2)$$

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

subject to

$$|\dot{\sigma}_i - \frac{1}{m}\sum_{j=1}^{m}\dot{\sigma}_j| \le \frac{\tau}{m}\sum_{j=1}^{m}\dot{\sigma}_j, \quad i = 1, \cdots, m;$$

$$T_L \le \boldsymbol{T}_i \le T_U, \quad i = 1, 2, \cdots, n;$$

$$Q_L \le \dot{Q} \le Q_U.$$

Equation (2) seeks to maximize the average deposition rate subject to the constraint that the difference between each individual sample point and the mean falls within some fraction τ of the mean. Again, the operating wall temperatures and flow rate are constrained to be within a given range; therefore bound constraints are prescribed for them. Equation (2) is inherently more difficult to solve than Equation (1) because we do not know what the mean value is a priori. Mathematically, this formulation gives rise to a nonlinearly-constrained optimization problem. The nonlinear constraints arise because L_∞-norm is used in the bounds for the deposition non-uniformity.

An alternative, which can reduce the number of nonlinear constraints by an order of magnitude in our case, is to use L_2-norm (that is, in the mean-squared-error sense) in Equation (2). The result is Equation (3) which has only one nonlinear constraint. A disadvantage for this simplification is that a few data points with large variations from the target may be accepted.

$$F(\boldsymbol{T}, \dot{Q}) = \min_{\boldsymbol{T}, \dot{Q}} -\frac{1}{m}\sum_{i=1}^{m}\dot{\sigma}_i \tag{3}$$

subject to

$$[\frac{1}{m}\sum_{i=1}^{m}(\dot{\sigma}_i - \frac{1}{m}\sum_{j=1}^{m}\dot{\sigma}_j)^2]^{1/2} \le \frac{\tau}{m}\sum_{j=1}^{m}\dot{\sigma}_j,$$

$$T_L \le \boldsymbol{T}_i \le T_U, \quad i = 1, 2, \cdots, n;$$

$$Q_L \le \dot{Q} \le Q_U.$$

Numerical results for Equation (2) and (3) will be given in later sections.

2.3 Maximizing Mean Deposition Rate Using Two Injectors

Suppose an additional injector is to be placed in the reactor. In addition, the location and flow rate of the second injector are also design parameters. The design objective is to predict the wall temperatures, the flow rates, and the position of the second injector such that the mean deposition rate is maximized while deposition non-uniformity is bounded by a given relative tolerance τ. If we also take into account that the current OVEND code allows the injectors to be put only at certain discrete locations (namely, midway between two adjacent wafers or at some pre-defined locations outside the range of the wafer stack), we have encountered a mixed integer programming problem which is much harder to solve than the nonlinearly constrained optimization problem. Moreover, the potential existence of many local minima in the solution space in this problem (as well as the previous one) means that different initial guesses may give different solutions.

Let D be the distance between the base of the reactor and Injector 2, and $\dot{Q}_1$ and $\dot{Q}_2$ be the flow rates of Injector 1 and 2, respectively. The optimization problem using L_∞-norm in the non-uniformity constraints can be expressed as

$$F(\boldsymbol{T}, \dot{Q}, D) = \min_{D \in S_D}\{\min_{\boldsymbol{T}, \dot{Q}_1, \dot{Q}_2} -\frac{1}{m}\sum_{i=1}^{m}\dot{\sigma}_i\} \tag{4}$$

subject to

$$|\dot{\sigma}_i - \frac{1}{m}\sum_{j=1}^{m}\dot{\sigma}_j| \le \frac{\tau}{m}\sum_{j=1}^{m}\dot{\sigma}_j, \quad i = 1, \cdots, m;$$

$$T_L \le \boldsymbol{T}_i \le T_U, \quad i = 1, 2, \cdots, n;$$

$$Q_{iL} \le \dot{Q}_i \le Q_{iU}, \quad i = 1, 2;$$

where S_D is a set of discrete values for D. For the test problem used in this work, the size of the set S_D is about 70. We will explore the space of S_D in our numerical experiments.

3 Software Architecture for Optimization with OVEND

A software design objective is to maintain a loosely-coupled computing environment for simulation (function evaluation) and optimization while allowing different configurations to be tested (by changing the input files and without recompiling the program). This is accomplished by a custom built interface between the optimization and the analysis code similar to the one described in [1]. The advantage of such an interface is that once it is in place, new optimization methods can be plugged in and tested with much ease. The optimization software chosen for our work is OPT++ which is an object-oriented library comprising many optimization algorithms. Since a

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

nonlinearly-constrained problem solver is currently not available in OPT++, we develop a link between OPT++ and a publicly available optimization package called NPSOL. In the following sub-sections, we describe each of the major components in the software structure as well as the interfaces between them. The software design is also depicted in Figure 2.

In Figure 2, the opt_ovend module sets up the optimization problem—reading in the problem dimension, initial guess, lower and upper bounds of the design parameters, etc.; instantiating an optimization algorithm from OPT++ (or linking NPSOL through OPT++); and providing the selected optimization algorithm with a function evaluator (and a constraint evaluator). When the function evaluator is called, it sets up the proper input files through the ifilter subroutine and then issues a UNIX system call to the analysis program OVEND. Upon termination, OVEND generates an output file containing the deposition rates which are read and processed by the ofilter subroutine and returned to opt_ovend to compute the function value. By using the ifilter and ofilter modules, OVEND is treated as a "black-box" by opt_ovend. This software structure allows modifications made to OVEND without having to modify or even re-compile the optimization portion of the code.

3.1 The OPT++ Optimization Library

The OPT++ software consists of a library of object-oriented optimization algorithms written in C++. Object-oriented programming emphasizes the creation and use of new data types formed from simple data types such as integer, double precision floating point number, character strings, etc. The more complex data type is specified as a *class*, which hides many implementation details and is visible only through a collection of data fields and class functions. For example, in defining a class for a certain optimization algorithm, the essential functions visible to the users are SetX (to define the initial values of the design variables), SetFcnAccrcy (to define the function precision), and optimize (to execute the optimization algorithm). With the definition of such an optimization class, the actual details of how optimization is performed are transparent to the users, and this is reasonable since users may only be concerned about the final results and not the methods to obtain them.

Another essential feature of object-oriented programming is the concept of inheritance. This is a powerful concept which helps to reduce the amount of coding. For example, nonlinear problems can be classified based on whether analytic derivatives are available or not. In OPT++, the class NLP0 defines a nonlinear problem which does not provide any analytic or finite-difference derivatives. If the first derivative is available, the nonlinear problem is classified as NLP1. The NLP1 class contains derivatives *in addition* to what is available in NLP0. Therefore, we can use the inheritance mechanism (namely, NLP1 inherits from NLP0) to define NLP1 such that all the data and functions available in NLP0 can be re-used. Similarly, NLP2 inherits from NLP1 and provides, in addition, second derivative information.

OPT++ provides a rich set of optimization methods—direct search method, nonlinear conjugate gradient method, Newton-like methods (Newton, Quasi-Newton, etc.), each in the form of a class. In addition, OPT++ version 1.5 also provides capabilities to handle problems with simple bound constraints on the design parameters. New methods continue to be added to the collection to handle more complicated problems.

3.2 The NPSOL Optimization Package

Equation (1) can readily be solved by a Gauss-Newton algorithm residing in OPT++. However, OPT++ is currently unable to solve nonlinearly constrained optimization problems represented by Equation (2) and (3). Therefore, we have developed an interface between OPT++ and a publicly available optimization package called NPSOL, which is a set of Fortran subroutines designed to minimize a smooth function subject to bounded, linear, and smooth nonlinear constraints. NPSOL [4] uses an iterative algorithm which solves a sequential quadratic programming subprogram at each iteration to find a search direction. In order to use the package, subroutines that define the objective and constraint functions and (optionally) their gradients must be provided.

3.3 Building an OPT++/NPSOL Interface

Since the current version of OPT++ (namely, version 1.5) cannot handle nonlinearly constrained problems, we have built an interface between OPT++ and NPSOL. The interface has been carefully designed to be re-used in OPT++ when nonlinearly constrained capabilities are added in the future. In the following we describe several software components for such an interface and the rationale for their inclusion.

1. As is with many software packages, the NPSOL optimizer demands a large number of parameters

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

to be passed to it (through calls to the parameter set-up subroutine and the optimizer) , and some of these parameters do not contain any useful information to the users (e.g. work arrays). It is therefore desirable to hide this data management from the users. As a result, two files called interface.c and npoptn.f are added to NPSOL to simplify the protocol between OPT++ and NPSOL. In addition, these files also serve as cross-language interface (between C/C++ and Fortran).

2. A new class (in the context of object-oriented programming) called OptNPSOL is defined which receives and stores parameters (data and function pointers) passed by the users and passes them on to the NPSOL optimizer. Furthermore, it provides an Optimize member function (as is with other optimizers in OPT++) that can be called to activate the optimizer.

3. A typical constraint evaluator returns all constraint values at each call, given a set of input values. It is not uncommon for the NPSOL optimizer, especially in the computation of finite-difference derivatives, to request a constraint evaluation using the same input values as with a previous request. In order to detect such occurrences and to avoid redundant computations, a data management module is designed to bridge this disparity. The added module stores the function values, constraint values, and (if provided) the gradient and Jacobian information returned by the function and constraint evaluators. Subsequent calls with the same parameter values will bypass the evaluators and instead fetch the requested information from the data bank, thus avoiding redundant evaluations.

4 Numerical Experiments

Several formulations of the CVD reactor design optimizations, as described earlier, are tested to demonstrate the capabilities of the OVEND/OPT++/NPSOL design tool. The reactor model has been given in Figure 1, which has an operating pressure of 0.6 torr, and where the reactor radius is 11.7 centimeters, the reactor length is 112 centimeters, the number of wafers in the stack is 50, and the wafer radius is 10 centimeters. The wafer spacing is set at 1.3 centimeters, with the first and last wafers located at 15.8 and 79.0 centimeters, respectively, from the base of the reactor. Both the heat conduction and the species transport models are discretized and solved with a hybrid Newton-time integration procedure embodied in the TWOPNT code developed at Sandia [11].

The first part of our experiment uses only one injector which is placed at a fixed location (9.75 centimeters from the base of the reactor). The second part uses two injectors with the first one at a fixed location and the position of the second injector as a design parameter. The temperature of the injectors are fixed at 300 K, and the chemical species injected consists of $Si(OC_2H_5)_4$ with a mole fraction of 1.0. We assume that on the reactor wall the temperature follows a linear profile between the first and the last wafer (i.e. between 15.8 and 79.0 centimeters from the base) and a constant profile above and below the wafer stack. The full temperature profile can therefore be specified by two temperatures at the locations 15.8 and 79.0 (which are T_1 and T_2 in the numerical examples given below).

4.1 Single Injector Test Case

In the first numerical example, the deposition rate is optimized according to the formulation given in Equation (2) where the constraints have been modified to be compatible with NPSOL, i.e.

$$F(\boldsymbol{T}, \dot{Q}) = \min_{\boldsymbol{T}, \dot{Q}} -\frac{1}{m} \sum_{i=1}^{m} \dot{\sigma}_i \qquad (5)$$

subject to

$$\begin{aligned} -\infty < \dot{\sigma}_i - \tfrac{1+\tau}{m} \textstyle\sum_{j=1}^{m} \dot{\sigma}_j \le 0, \quad i = 1, 2, \cdots, m; \\ 0 \le \dot{\sigma}_i - \tfrac{1-\tau}{m} \textstyle\sum_{j=1}^{m} \dot{\sigma}_j < \infty, \quad i = 1, 2, \cdots, m; \\ T_L \le \boldsymbol{T}_i \le T_U, \quad i = 1, 2, \cdots, n; \\ Q_L \le \dot{Q} \le Q_U \end{aligned}$$

with the parameters given by

n = 2 (number of wall temperatures)
m = 400 (number of deposition rates)
T_L = 948 K
T_U = 1023 K
Q_L = 25 sccm
Q_U = 125 sccm
h = 10^{-5} (relative finite difference interval).

Different non-uniformity tolerances, τ, and initial guesses are tested. Furthermore, both the L_∞-norm and L_2-norm for the deposition non-uniformity constraints are tested.

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

4.1.1 Numerical Results

Numerical results show that the initial guess for the temperature profile has little effect on the converged mean deposition rate. Different initial flow rates $\dot{Q}$, on the other hand, may lead to different answers (or non-convergence). We therefore perform the tests with fixed initial temperatures at $T_1 = T_2 = 950$ K and vary the initial $\dot{Q}$ and τ. The results are given in Table 1 and 2 for L_∞- and L_2-norm, respectively (data in Table 2 are obtained for an initial $\dot{Q}$ of 30.0 sccm). The achievable mean deposition rates with respect to τ are also plotted in Figure 3.

4.1.2 Observations and Discussion

A few observations were gathered during numerical simulation and from the numerical results :

1. For the L_∞-norm case, we found that with the given bounds for the wall temperatures and flow rate, the smallest τ for a successful run (such that a feasible solution is found) is about 0.024, meaning that one cannot achieve better than about 2.4 % variation on the deposition rates using 1 injector and a linear temperature profile. The reason is that for $\tau < 0.024$, T_1 and $\dot{Q}$ must drop below their corresponding lower bounds in order to achieve our design objective.

2. OVEND was initially set up to run at a convergence tolerance of 10^{-9}. We found that this is too noisy to generate, in **Get_F**, finite-difference derivatives with respect to the flow rate using a relative finite difference step size of 10^{-5} or smaller. To improve robustness, we decrease the convergence tolerance to 10^{-11}. Little effect in the computer run-time was observed as a result of this change.

3. From Figure 3 and 4, the mean deposition rates and τ appear to exhibit a linear relationship in the log-log scale. The slope of the line is approximately equal to 2 in both cases. Hence, we can approximate the achievable mean deposition rate by $\bar{\dot{\sigma}} \approx c\tau^2$ where c is about 8.6×10^4 and 4×10^5 for the L_∞-norm and L_2-norm cases, respectively.

4.2 Two-Injector Test Case

This numerical example uses two injectors with Injector 1 placed at a fixed position of 9.75 cm from the base of the reactor and Injector 2 placed (manually) at various allowable (by the OVEND analysis program) locations. We solve Equation (4) with its nonlinear constraints modified to be compatible with NPSOL, i.e.,

$$\min_{D \in S_D} \{ \min_{\boldsymbol{T}, \dot{Q}_1, \dot{Q}_2} -\frac{1}{m} \sum_{i=1}^{m} \dot{\sigma}_i \} \tag{6}$$

subject to

$$\begin{aligned}
-\infty < \dot{\sigma}_i - \tfrac{1+\tau}{m} \textstyle\sum_{j=1}^{m} \dot{\sigma}_j \le 0, \quad i = 1, 2, \cdots, m; \\
0 \le \dot{\sigma}_i - \tfrac{1-\tau}{m} \textstyle\sum_{j=1}^{m} \dot{\sigma}_j < \infty, \quad i = 1, 2, \cdots, m; \\
T_L \le \boldsymbol{T}_i \le T_U, \quad i = 1, 2, \cdots, n; \\
Q_{iL} \le \dot{Q}_i \le Q_{iU} \quad i = 1, 2;
\end{aligned}$$

with parameters given by

$$\begin{aligned}
n &= 2 \\
m &= 400 \\
T_L &= 948 \text{ K} \\
T_U &= 1023 \text{ K} \\
Q_{1L} &= 25 \text{ sccm} \\
Q_{1U} &= 125 \text{ sccm} \\
Q_{2L} &= 0 \text{ sccm} \\
Q_{2U} &= 125 \text{ sccm} \\
\tau &= 0.03 \text{ (0.015 for } L_2\text{-norm)}.
\end{aligned}$$

The initial wall temperatures are 950 K and the initial flow rates are 30 and 0 sccm, respectively. Instead of using mixed integer programming solvers, we perform the outer minimization in Equation (6) by running a suite of test cases with different Injector 2 positions and hand-picking the optimal solution.

4.2.1 Numerical Results

Tables 3 and 4 show the achievable mean deposition rates for different locations of injector 2, using L_∞ and L_2 norms, respectively, for the nonlinear constraints. The plots of achievable mean deposition rates with respect to D are also given in Figure 4.

4.2.2 Observations and Discussion

It can be observed that adding a second injector to the system in general increases the achievable mean deposition rates (see Table 3 and 4). The largest improvement is obtained when the second injector is located at about 22 to 24 cm above the reactor base. However, the presence of the second injector gives no improvement if it is placed at 25 to 64 cm from the reactor base. The best achievable deposition rates are 132 and 112 for using L_∞ and L_2 norms, respectively (recall that these two values are obtained from using two

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

different τ's). The last column of Table 4 indicates the position of the wafer where maximum deviation from the mean is found. The first number is the wafer number and the second is the radial point number. For example, for $D = 16.5$, the maximum deviation is at the second wafer (from the bottom) and the eighth radial point, i.e. the point closest to Injector 2.

Achieving higher mean deposition rates often comes with higher flow rate requirement for the injectors. In Table 3 and 4 we also include a column showing the total flow rate requirements, which may be used by designers to decide whether design parameters other than the optimal should be considered.

5 Summary

In this report, we have shown how to formulate an optimal optimization problem for the design of CVD reactors with the objective to achieve maximum deposition without sacrificing production yield. Several formulations using the non-uniformity constraints are presented and their computational needs are examined. Moreover, we have described a detailed design and implementation of the program to solve the optimization problems. Numerical results are also given to show its effectiveness.

References

[1] C. D. Moen, P. A. Spence, and J. C. Meza, *Optimal Heat Transfer Design of Chemical Vapor Deposition Reactors,* Report SAND95-8223, Sandia National Laboratories, Livermore, California, April 1995.

[2] J. C. Meza, T. D. Plantenga, *Optimal Control of a CVD Reactor for Prescribed Temperature Behavior,* Report SAND95-8224, Sandia National Laboratories, California, April 1995.

[3] C. D. Moen, P. A. Spence, J. C. Meza, and T. D. Plantenga, *Automatic Differentiation for Gradient-Based Optimization of Radiatively Heated Microelectronics Manufacturing Equipment,* AIAA Paper 96-4118, Sixth AIAA/NASA/ISSMO Symposium on Multidisciplinary Analysis and Optimization, Bellevue, Washington, Sept. 1996.

[4] P. E. Gill, W. Murray, M. A. Saunders, and M. H. Wright, *User's Guide for NPSOL (Version 4.0) : A Fortran Package for Nonlinear Programming,* Technical Report SOL 86-2, Department of Operations Research, Stanford University, Jan. 1986.

[5] J. F. Grcar, W. G. Houf, W. G. Breiland, *A Model for Low Pressure Chemical Vapor Deposition in a Hot-wall Tubular Reactor,* Materials Science and Engineering B, Solid State Materials for Advanced Technology, Vol. 17, No. 1/3, 1993, pp. 163-171.

[6] J. C. Meza, *OPT++ : An Object-Oriented Class Library for Nonlinear Optimization,* Report SAND94-8225, Sandia National Laboratories, Livermore, California, March 1994.

[7] A. Dip, *Fast Thermal Processing : Batch comes back,* Solid State Technology, Vol. 39, No. 6, 1996, pp. 113-124.

[8] R. J. Kee, G. Dixon-Lewis, J. Warnatz, M. E. Coltrin, and J. A. Miller, *A Fortran Computer Code Package for the Evaluation of Gas-phase Multicomponent Transport Properties,* Report SAND86-8246, Sandia National Laboratories, Livermore, California, 1991.

[9] R. J. Kee, F. M. Rupley, and J. A. Miller, *Chemkin-II : A Fortran Chemical Kinetics Code Package for the Analysis of Gas-phase Chemical Kinetics,* Report SAND89-8209, Sandia National Laboratories, Livermore, California, 1991.

[10] M. E. Coltrin, R. J. Kee, and F. M. Rupley, *Surface Chemkin : A Fortran Package for Analyzing heterogeneous Chemical Kinetics at a solid-surface-gas-phase Interface,* Report SAND90-8003B, Sandia National Laboratories, Livermore, California, 1991.

[11] J. F. Grcar, *The* TWOPNT *Program for Boundary Value Problems,* Report SAND91-8230, Sandia National Laboratories, Livermore, California, 1992.

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

Table 1: **Numerical Results for Single Injector Case (L_∞-norm case)**

Initial data		Optimized design parameters			Output Data	
initial $Q(sccm)$	τ	T_1 (K)	T_2 (K)	Q (sccm)	$\bar{\bar{\sigma}}$ (Å/min)	# feval
30.0	0.024	948.1	981.4	27.8	47.78	36
30.0	0.026	954.0	986.8	33.5	56.20	66
30.0	0.028	959.9	992.4	40.3	66.11	68
30.0	0.030	965.5	997.7	48.3	77.09	100
30.0	0.032	970.7	1002.3	57.7	88.82	100
30.0	0.034	975.4	1007.1	65.9	100.30	120
30.0	0.036	980.0	1012.3	72.9	112.22	164
30.0	0.038	984.3	1016.3	82.7	124.98	193
30.0	0.040	988.4	1021.0	90.6	138.19	176
Last column : number of function evaluations needed.						

Table 2: **Numerical Results for Single Injector Case (L_2-norm case)**

Input	Optimized design parameters			Output Data	
τ	T_1 (K)	T_2 (K)	Q (sccm)	$\bar{\bar{\sigma}}$ (Å/min)	# feval
0.012	954.5	982.7	38.8	57.44	160
0.014	965.4	993.3	53.9	76.91	506
0.016	974.6	1002.3	70.8	97.76	144
0.018	982.4	1009.4	90.7	119.55	122
0.020	989.1	1014.0	121.2	142.71	136
Last column : number of function evaluations needed.					

Table 3: **Numerical Results for Two-Injector Case A**

D (cm)	T_1 (K)	T_2 (K)	Q_1 (sccm)	Q_2 (sccm)	Total Q	$\bar{\bar{\sigma}}$ (Å/min)
16.455	970.53	1001.37	33.92	24.45	58.37	88.37
19.033	980.47	1009.53	61.82	18.45	80.27	113.58
22.902	986.50	1013.98	78.57	19.92	98.49	131.62
24.191	982.98	1010.26	70.31	18.02	88.33	119.89
30.638	966.59	999.25	46.61	1.27	47.88	78.42
31.927	965.80	998.19	47.60	0.38	47.98	77.36
62.873	965.53	997.71	48.30	0.00	48.30	77.09
64.162	965.88	998.41	47.98	0.15	48.13	77.61
66.741	968.05	1001.86	48.47	0.38	48.85	81.52
69.320	969.76	1004.82	48.44	0.81	49.25	84.65
71.898	971.57	1007.75	49.03	1.20	50.23	88.22
77.056	968.69	1002.84	48.82	0.89	49.71	82.79
81.851	967.31	1000.52	48.73	0.89	49.62	80.29
Last column : location (wafer and grid point number) of maximum deviation.						

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

Table 4: **Numerical Results for Two-Injector Case B**

D (cm)	T_1 (K)	T_2 (K)	Q_1 (sccm)	Q_2 (sccm)	Total Q	$\bar{\sigma}$ (Å/min)	Max at
16.455	970.71	998.92	25.00	37.35	62.35	88.53	(2, 8)
19.033	976.19	1002.29	43.79	35.10	78.89	102.48	(4, 8)
24.191	979.14	999.95	81.39	29.30	110.69	112.44	(8, 8)
25.480	978.87	999.03	86.17	27.57	113.74	111.87	(1, 8)
30.638	974.68	995.16	82.60	14.87	97.47	99.97	(1, 8)
31.927	971.52	988.67	95.97	18.34	114.31	94.58	(1, 8)
33.217	971.93	995.02	71.03	7.05	78.08	92.02	(1, 8)
34.506	970.88	995.80	66.13	3.78	69.91	89.07	(1, 8)
35.796	970.29	997.09	63.01	1.21	64.22	87.47	(1, 8)
37.085	970.19	997.99	62.07	0.00	62.07	87.21	(1, 8)
66.741	970.19	997.99	62.06	0.00	62.06	87.21	(1, 8)
69.320	970.42	998.92	60.23	0.27	60.50	87.28	(1, 8)
71.898	970.80	1000.16	58.53	0.57	59.10	87.69	(1, 8)
77.056	970.32	998.49	61.17	0.20	61.37	87.29	(1, 8)
81.851	970.19	998.00	62.06	0.00	62.06	87.21	(1, 8)

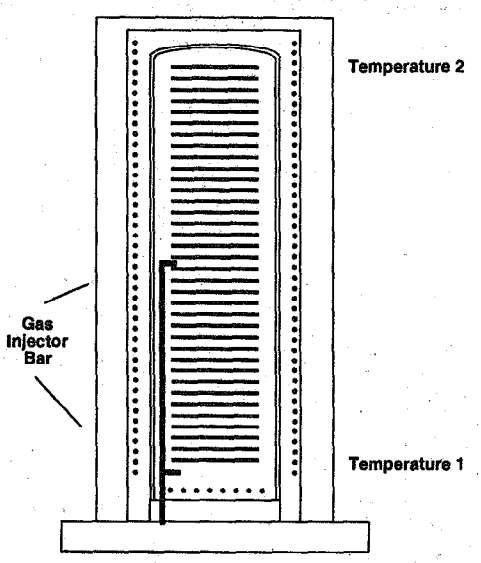

Figure 1: **Schematic diagram of a CVD reactor**

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE

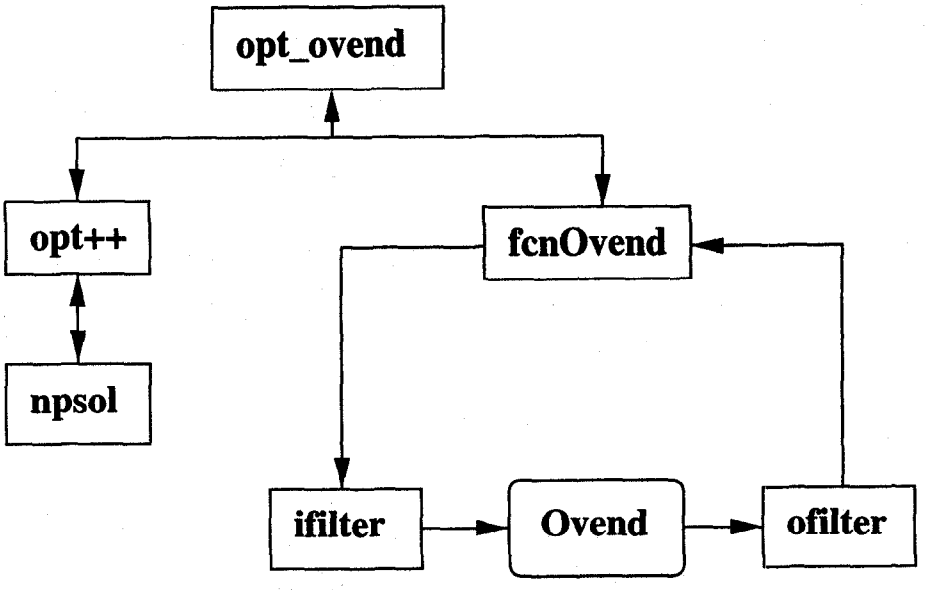

Figure 2: **Block Diagram for OVEND/OPT++/NPSOL**

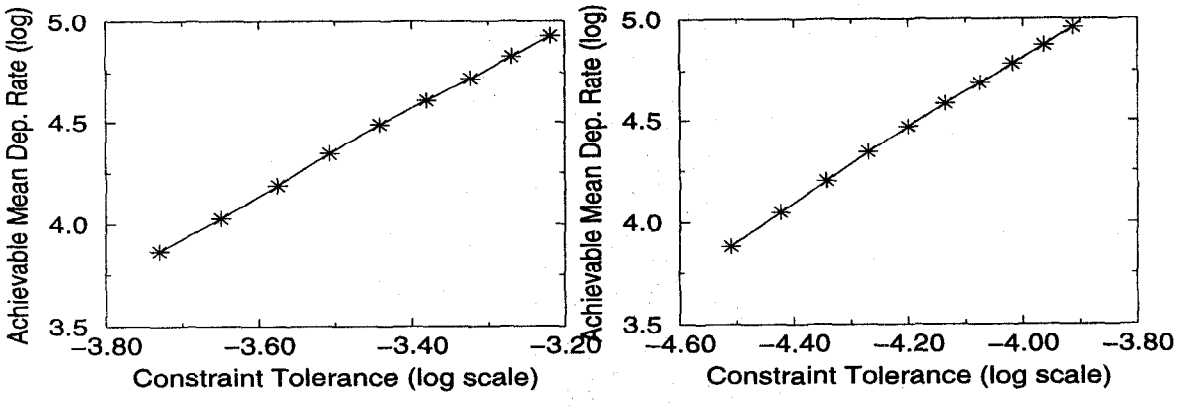

Figure 3: **Mean deposition rates as a function of τ : using (left) L_∞-norm, and (right) L_2-norm in the nonlinear constraints**

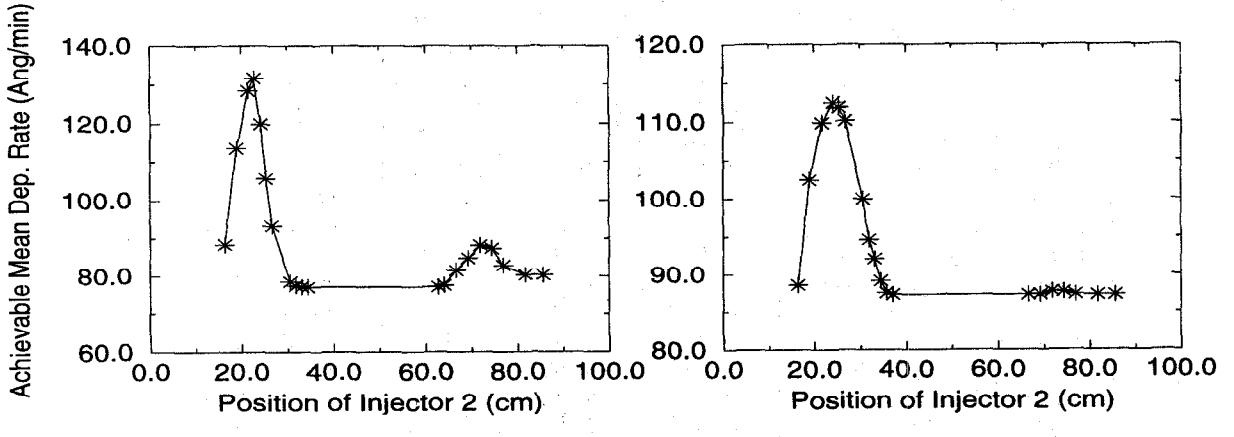

Figure 4: **Mean deposition rates as a function of Injector 2 position : using (left) L_∞-norm, and (right) L_2-norm in the nonlinear constraints**

Proceedings of the 30th Annual Simulation Symposium (SS '97)
1080-241X/97 $10.00 © 1997 IEEE